

(51) Int.Cl.⁶

識別記号

F I

G 0 6 T 15/00
11/00

G 0 6 F 15/72

4 5 0 A
3 5 0

審査請求 未請求 請求項の数 4 O L (全 10 頁)

(21) 出願番号

特願平9-355995

(22) 出願日

平成9年(1997)12月25日

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番
1号

(72) 発明者 大塚 竜志

神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(74) 代理人 弁理士 小笠原 吉義 (外2名)

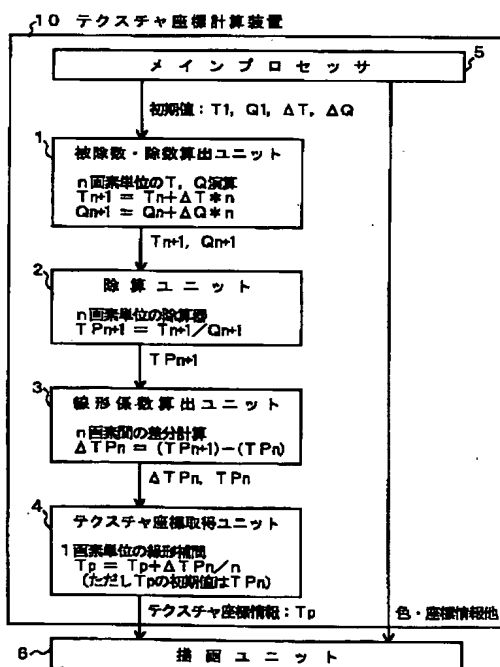
(54) 【発明の名称】 3次元コンピュータグラフィックスのテクスチャマッピング座標計算装置および方法

(57) 【要約】

【課題】 3次元コンピュータグラフィックスのテクスチャマッピング座標計算装置および方法に関し、テクスチャデータの貼り付けにあたって奥行きの変形補正に必要となる除算処理量を軽減することを目的としている。

【解決手段】 被除数・除数算出ユニット1は、画素aにn画素先行した画素bのテクスチャ座標を求めるための被除数・除数を算出し、除算ユニット2は、この値から除算により画素bのテクスチャ座標を求める。線形係数算出ユニット3は、画素a、bのテクスチャ座標の差分から線形補間係数を算出する。テクスチャ座標取得ユニット4は、画素aのテクスチャ座標と線形補間係数とから画素a、b間の各画素のテクスチャ座標を算出する。

本発明の原理説明図



【特許請求の範囲】

【請求項 1】 3次元コンピュータグラフィックスのテクスチャマッピング座標を算出する装置であって、描画すべき画素(a)のテクスチャ座標と、前記描画すべき画素(a)に対しn画素先行した画素(b)におけるテクスチャ座標を求め、2つのテクスチャ座標の差分から、前記画素(a)と前記画素(b)の間の画素についてのテクスチャ座標の線形補間係数を算出する第1の手段と、前記描画すべき画素(a)のテクスチャ座標と前記線形補間係数とから、前記画素(a)と前記画素(b)の間の各々の画素におけるテクスチャ座標を求める第2の手段とを備えることを特徴とする3次元コンピュータグラフィックスのテクスチャマッピング座標計算装置。

【請求項 2】 請求項1に記載する3次元コンピュータグラフィックスのテクスチャマッピング座標計算装置であって、前記第1の手段は、前記n画素先行した画素(b)のテクスチャ座標を除算により求めるためのn画素単位の被除数と除数を算出する被除数・除数算出部と、求めた値により除算を行い、前記画素(b)のテクスチャ座標を算出する除算部と、前記画素(a)のテクスチャ座標と前記画素(b)のテクスチャ座標との差分から、前記画素(a)と前記画素(b)の間の画素についてのテクスチャ座標の線形補間係数を算出する線形係数算出部とから構成され、前記第2の手段は、前記画素(a)のテクスチャ座標に前記線形補間係数を加算して、前記画素(a)と前記画素(b)の間の各々の画素におけるテクスチャ座標を求めるテクスチャ座標取得部とから構成されることを特徴とする3次元コンピュータグラフィックスのテクスチャマッピング座標計算装置。

【請求項 3】 請求項2に記載する3次元コンピュータグラフィックスのテクスチャマッピング座標計算装置であって、前記テクスチャ座標取得部は、前記画素(a)のテクスチャ座標にそれぞれ前記線形補間係数をk倍($k=0, 1, 2, \dots, n-1$)して加算する演算を並列に実行する手段を持ち、前記画素(a)と前記画素(b)の間の各々の画素のテクスチャ座標を同時に算出することを特徴とする3次元コンピュータグラフィックスのテクスチャマッピング座標計算装置。

【請求項 4】 3次元コンピュータグラフィックスのテクスチャマッピング座標を算出する方法であって、描画すべき画素(a)のテクスチャ座標と、前記描画すべき画素(a)に対しn画素先行した画素(b)におけるテクスチャ座標を求め、2つのテクスチャ座標の差分から、前記画素(a)と前記画素(b)の間の画素についてのテクスチャ座標の線形補間係数を算出し、前記描画すべき画素(a)のテクスチャ座標と前記線形補間係数とから、前記画素(a)と前記画素(b)の間の各々の画素におけるテクスチャ座標を求めることを特徴とする3次元コンピュータグラフィックスのテクスチャマッピング座標計算方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、3次元コンピュータグラフィックスの高速描画を実現する技術に係り、特に3次元コンピュータグラフィックスのテクスチャマッピングに必要となるテクスチャ座標を、比較的簡易な構成により高速に計算する装置およびその方法に関する。

【0002】近年、3次元コンピュータグラフィックスは、従来の主要用途であるコンピュータデザインの可視化にとどまらず、ゲームの分野などにも多く利用されるようになってきた。これらの分野では、ユーザーの入力に対し、リアルタイムで反応するために、高速な3次元コンピュータグラフィックスの処理能力が必要とされている。

【0003】このためには、少ない形状データでリアリティの高い画像を生成する必要があり、テクスチャマッピングと呼ばれる技術が使用されている。テクスチャマッピングは、あらかじめテクスチャデータと呼ばれる2次元画像を用意し、ポリゴンの描画時の画素単位に、このテクスチャデータを参照してポリゴンの画素情報にテクスチャデータを反映させる処理である。これにより、少ないポリゴンでも複雑なテクスチャデータを貼り付けることにより、リアリティの高い画像を生成することができる。

【0004】

【従来の技術】まず、3次元コンピュータグラフィックスの一般的なシステムの全体構成について説明する。

【0005】図6は、3次元コンピュータグラフィックスの一般的なシステムの全体構成を示す図である。メインプロセッサ90は、3次元コンピュータグラフィックスの実際のアプリケーションを実行するプロセッサで、3次元情報が管理されている。例えば、機械系CADなどでは、車のボディの形状等のデータが、建築系のCADなどではビルなどの形状などのデータが3次元情報として管理されている。

【0006】これらのデータは、記憶装置上では単なる座標データ列などにすぎないため、対象が車なら車のように、ビルならビルのように、これらのデータをもとにして人間が実物を見るような画像を作りだす(可視化する)のが、3次元コンピュータグラフィックス処理(3次元CG処理)である。

【0007】一般的に、メインプロセッサ90のアプリケーションが管理する3次元情報の内部形式は、各アプリケーションにより様々であるが、3次元CG処理の構成を容易にするため、3次元CG処理が受け付ける3次元形状データは、ある程度パターン化されている。

【0008】その一つとして、図7に示すような3次元情報を複数の多面体で近似して表現する手法がある。図7(A)に示すように、自動車のボディは、実際には、50 多角形の組み合わせで近似して表現することができる。

この例では、矩形の組み合わせで自動車のボディを表現しているが、システムによって様々な表現方法をとることが可能である。一般的には、矩形やさらに複雑な多角形も、複数の三角形を組み合わせることで表現できるため、三角形を組み合わせることで表現するシステムが多い。

【0009】以下、説明の簡略化のために、三角形を使用した表現について説明する。三角形は、図7(B)に示すように点A、B、Cの頂点があり、3次元コンピュータグラフィックスでは、この3つの頂点における3次元座標、色、法線、テクスチャ座標などの情報が、3次元形状データとして幾何変換部92に渡される。

【0010】幾何変換部92は、これらの情報に加え、視点や光源、面の材質情報などを用いて、実際に表示されるべき画面に合わせた次のようなポリゴンデータと呼ぶ情報を算出する。

【0011】1)画面上でのX・Y座標と、前後関係を判定し隠れ面消去を実現するための奥行き情報

2)光源からの照光具合による各頂点の色情報

3)テクスチャ座標情報

なお、前処理プロセッサ91の処理をメインプロセッサ90で実行する場合もある。

【0012】ポリゴンデータは、各頂点における情報であるが、これから、ポリゴン内部を塗りつぶすための情報にブレイクダウンする必要がある。このブレイクダウンの処理を行うのがスパン分解部93とフラグメント生成部94である。

【0013】スパン分解部93では、図8(A)に示すような頂点で構成されるポリゴンデータを、画面の水平方向に分割し、水平方向の描画開始点と増分、終了点の情報からなる図8(B)に示すスパンデータを生成する。

【0014】フラグメント生成部94は、このスパンデータをもとに、図8(C)に示すフラグメントデータと呼ばれる1画素単位のX・Y座標、奥行き情報、色情報、テクスチャ情報などを生成する。

【0015】画素描画部95は、このフラグメントデータをもとに、奥行き情報による描画点の前後関係の判定(隠れ面処理)や、テクスチャ座標からテクスチャマッピングデータの読み込みや色情報の計算などを行い、最終的な描画点の画素の色などを決定する。一般的には、コンピュータグラフィックスの場合、色の3原色赤緑青の頭文字をとってRGB値と呼ばれる値をフレームメモリ97に書き込む。

【0016】さらに、色の透過情報としてアルファ値と呼ばれる値を使用する場合もあり、この場合RGBA値として扱う。表示制御部98は、このフレームメモリ97の値を1秒間に60回程度読み出し、CRT等の表示装置99に表示する。

【0017】前処理プロセッサ91が扱う3次元形状データは、頂点情報の他、視点や光源情報などの多くの情

報を含むうえ、アプリケーションとの関係で多様な形式であるため、ここでは省略して、どのシステムでも大差のないポリゴンデータ以下のデータについて説明する。

【0018】図8(A)に示すポリゴンデータは、3つの頂点A、B、Cについて、X・Y座標、奥行きZ情報、色RGBA情報、テクスチャ座標情報S、Tおよび奥行きの歪み補正の情報Qを持つ。

【0019】頂点Aでのこれらの情報は、次のように表す。

「Xa, Ya, Za, Ra, Ga, Ba, Aa, Sa, Ta, Qa」

同様に、頂点B、Cについては、それぞれ次のように表す。

【0020】

「Xb, Yb, Zb, Rb, Gb, Bb, Ab, Sb, Tb, Qb」

「Xc, Yc, Zc, Rc, Gc, Bc, Ac, Sc, Tc, Qc」

図8(B)に示すスパンデータは、3つの頂点A、B、Cからなる三角形を水平方向に分解した、各々の水平線での開始点と増分および終了点の情報である。開始点でのX・Y座標、奥行きZ情報、色RGBA情報、テクスチャ座標情報S、Tおよび奥行きの歪み補正情報Qおよびそれらの増分の各々の情報を、次のように表す。

【0021】

「Xs, Ys, Zs, Rs, Gs, Bs, As, Ss, Ts, Qs」

「dXs, dYs, dZs, dRs, dGs, dBs, dAs, dSs, dTs, dQs」

図8(B)に示すように頂点が並んでいたとすると、辺AB上に開始点がある場合には、Y座標Yiでのスパン情報は、次のような計算で求まる。

【0022】 $M_s(Y_i) = (M_b - M_a) * (Y_i - Y_a) / (Y_b - Y_a)$

ただし、 $Y_i = Y_a \sim Y_b$ 、Mは、X、Z、R、G、B、A、S、T、Qの各々を表す。

【0023】なお、計算式中に $(Y_b - Y_a)$ により除算があるが、これはYiの変化によらず一定であるため、先に逆数 $1 / (Y_b - Y_a)$ を計算しておくことで乗算に置き換えることができる。

【0024】増分は、次のように計算できる。

$dM_s = ((Y_b - Y_a) * (M_c - M_a) - M_b(Y_c - Y_a)) / ((Y_b - Y_a) * (X_c - X_a) - X_b(Y_c - Y_a))$

増分はYiによらず一定であるため、最初に1回計算するだけでよい。終了点はX座標だけで済む。これをXeとすると、次のように計算できる。

【0025】 $X_e(Y_i) = (X_c - X_a) * (Y_i - Y_a) / (Y_c - Y_a)$

図8(C)に示すフラグメントデータは、スパン上の1画素ずつの情報で、開始点と増分から算出することができる。

【0026】各画素のX・Y座標、奥行きZ情報、色RGBA情報、テクスチャ座標情報S、Tおよび奥行きの歪み補正の情報Qおよび増分の各々の情報を次のように表す。

【0027】「 $X_p, Y_p, Z_p, R_p, G_p, B_p, A_p, S_p, T_p, Q_p$ 」
 スパン X_s 、 X_e 上での開始点の各情報と増分を M_s 、
 dM_s (M は Z, R, G, B, A, S, T, Q の各々)と
 すれば、スパン上の X 座標 X_i のフラグメント情報は次の
 ように計算できる。

$$\text{【0028】 } M_f(X_i) = M_s + dM_s * (X_i - X_s) / (X_e - X_s)$$

ただし、 $X_i = X_s \sim X_e$ 、 M は、 Z, R, G, B, A, S, T, Q の各々を表す。

【0029】このようにして、フラグメントデータにより
 画素描画部95は、実際の描画処理を開始するが、テ
 クスチャ座標 S_f, T_f をそのままテクスチャメモリのア
 クセスに適用すると、奥行きに歪みを持っているため不
 自然な画像となる。これを避けるため、画素描画部95
 では、 $S_f/Q_f, T_f/Q_f$ を計算して、奥行きの歪
 み補正を行う必要がある。

【0030】なお、 $RGBA$ 値についても厳密には歪み
 が発生するが、色情報の変化は滑らかであるため、歪み
 を知覚することはほとんどなく、歪み補正を省略するこ
 とができる。

【0031】この奥行きの歪み補正について、具体例に
 従ってさらに詳しく説明する。3次元コンピュータグラ
 フィックスでは、遠近感を表現するために、視点から離
 れるに従って、小さく描くようにしている。

【0032】図9は、市松模様のテクスチャを矩形に貼
 り付け、矩形の縦中央線を中心として回転させた様子を
 示している。図9(A)に示す図では、矩形は視線に対
 し直交しているが、図9(B)～(D)に従い、矩形は
 だんだん傾いて表示されている。このように、3次元コ
 ンピュータグラフィックスでは、遠近感に合わせて、テ
 クスチャ上の市松模様も手前は大きく、遠方では縮んで
 表される必要がある。

【0033】このため、図10(A)に示すように視線
 に直交するポリゴンデータを描画する際には、隣り合う
 画素に対応するテクスチャデータ上の画素間の距離は同
 じであるが、図10(B)に示すように視線に直交しな
 いポリゴンデータを描画する際には、手前側(表示面左
 側)と奥側(表示面右側)では、隣り合う画素に対応す
 るテクスチャデータ上の画素間の距離が異なるため、補
 正が必要になる。

【0034】すなわち、例えば図11(A)に示すよう
 なテクスチャデータを、図11(B)に示すような視線
 と直交しない矩形にマッピングをする場合には、対応す
 るテクスチャデータの範囲が、奥側(表示面右側)にな
 るにしたがって、図11(C)に示すように広がっていく
 ので、ポリゴンデータにマッピングされるテクスチャ
 座標の補正が必要となる。

【0035】図10(B)において、点Aと点Bの画素
 座標が X_a, X_b (本来ディスプレイ上の座標は2次元
 で表されるが説明の簡略化のため Y 座標を省略して1要

素とする)、奥行きが Z_a, Z_b 、テクスチャデータ上
 の座標が T_a と T_b というように対応する場合(本来2
 次元のテクスチャ座標は2次元で表されるが説明の簡略
 化のため1要素とする)、点Aと点Bの間の座標 X_p の
 画素に対するテクスチャ座標 T_p は、次のように表され
 る。

$$\text{【0036】 } <\text{式1}> T_p = (T_1 + \Delta T * (X_p - X_a)) / (Q_1 + \Delta Q * (X_p - X_a))$$

これらの補正処理を、点Aから点Bの間の各画素ごと
 に行う必要がある。

【0037】ここで、

$$T_1 = T_a / Z_a$$

$$T_2 = T_b / Z_b$$

$$\Delta T = (T_2 - T_1) / (X_b - X_a)$$

$$Q_1 = 1 / Z_a$$

$$Q_2 = 1 / Z_b$$

$$\Delta Q = (Q_2 - Q_1) / (X_b - X_a)$$

であり、これらの値は、点Aから点Bの間の点につい
 ては一定である。

20 【0038】

【発明が解決しようとする課題】前述の<式1>から判
 るように、テクスチャ座標 T_p を求める処理には、除算
 が必要となる。通常、除算演算には逐次演算処理が必要
 となる。1画素あたりの描画処理の時間が逐次処理で除
 算結果を求めるまでの時間より短い場合には、除算演算
 がボトルネックとならないよう、除算演算部をパイプラ
 イン化する必要があり、ハードウェア規模が増大する。

30 【0039】また、複数の描画プロセッサを用いて、画
 素ごとに並列処理を行うためには、個々の描画プロセッ
 サに除算演算のハードウェアが必要となる。一方、簡易
 なシステムでは、テクスチャの補正処理そのものを省略
 する選択もあるが、図12に示すように、テクスチャに
 遠近感が反映されないため、画像は、奥行きの表現が不
 自然で違和感が大きいものとなる。

【0040】本発明は、テクスチャデータの貼り付けに
 あたって奥行きの歪み補正に必要な除算処理量を軽減
 することを目的としている。

【0041】

40 【課題を解決するための手段】本発明では、テクスチャ
 座標を取得する際に、1画素単位で除算する代わりに n
 画素単位で除算を行い、その画素間の1画素ずつのテク
 スチャ座標値は線形補間により求めるようにする。

【0042】従来、図6に示すフラグメント生成部94
 側において $X_s \sim X_e$ の間で1画素ずつ生成し、画素描
 画部95でテクスチャメモリへのアクセスのために $S,$
 T, Q を計算していたのに対し、本発明では、これらの
 処理を分離し、フラグメント生成部94側は複数画素分
 を算出し、以下で説明するテクスチャ座標計算装置により
 除算を行い、計算した画素間の1画素ずつを線形補間
 50 で求めて、描画ユニット(画素描画部)に渡す。描画ユ

ニットでは、もはや除算は必要ない。他のZ, R, G, B, A情報に関しては、歪みの影響がないか、もしくは無視できるほど小さいため、フラグメント生成部側で1画素ずつ計算した値を画素の描画ユニットに入れる。

【0043】図1は、本発明の原理説明図である。テキスト座標計算装置10は、被除数・除数算出ユニット1、除算ユニット2、線形係数算出ユニット3、テキスト座標取得ユニット4、メインプロセッサ5からなる。描画ユニット6は、テキスト座標計算装置10が算出したテキスト座標をもとに、テキストデータが格納されているテキストメモリにアクセスし、テキストデータを貼り付けて描画する。

【0044】なお、以下では、図10(B)において説明したように、点Aと点Bの画素座標を X_a, X_b 、奥行きを Z_a, Z_b 、テキストデータ上の座標を T_a と T_b として説明する。

$$\begin{aligned} T1 &= T_a / Z_a \\ T2 &= T_b / Z_b \\ \Delta T &= (T2 - T1) / (X_b - X_a) \\ Q1 &= 1 / Z_a \\ Q2 &= 1 / Z_b \\ \Delta Q &= (Q2 - Q1) / (X_b - X_a) \end{aligned}$$

とする。なお、本来2次元のテキスト座標は、2次元で表されるが説明の簡略化のため1要素として説明する。

【0046】被除数・除数算出ユニット1は、除算ユニット2で用いる被除数・除数を算出するユニットである。被除数・除数算出ユニット1は、 $T1, Q1$ を初期値とし、 $\Delta T, \Delta Q$ を n 倍した値を増分値として足し込んでいく。すなわち、次の演算を行う。

$$\begin{aligned} T_{n+1} &= T_n + \Delta T * n \\ Q_{n+1} &= Q_n + \Delta Q * n \end{aligned}$$

なお、ここで、 $T1, Q1, \Delta T, \Delta Q$ などの値は、点Aから点Bまでの間は一定であるため、この間の画素の処理が始まる前に、あらかじめ計算しておく。この計算は、例えばメインプロセッサ5で行ってもよいし、メインプロセッサ5と図1中では図示しないが被除数・除数算出ユニット1の間に前処理プロセッサを置き、ここで処理するようにしてもよい。

【0048】除算ユニット2は、被除数・除数算出ユニット1により算出した被除数 T_{n+1} と除数 Q_{n+1} とから、除算結果 TP_{n+1} である n 画素ごとのテキスト座標を算出するユニットである。次の演算を行う。

$$TP_{n+1} = T_{n+1} / Q_{n+1}$$

線形係数算出ユニット3は、 n 画素単位で得られる除算結果 TP_{n+1} と1つ前の除算結果 TP_n とから、それらの差分 ΔTP_n を算出するユニットである。次の演算を行う。

$$\Delta TP_n = (TP_{n+1}) - (TP_n)$$

テキスト座標取得ユニット4は、1画素単位のテクス

チャ座標 TP を計算するユニットである。テキスト座標取得ユニット4は、除算結果 TP_n を初期値とし、線形補間係数として $\Delta TP_n / n$ を足し込んでいく。すなわち、次の演算を行う。

$$TP = TP + \Delta TP_n / n \quad (\text{ただし、} TP \text{の初期値は} TP_n)$$

なお、 n を2の累乗とすることで、 $\Delta TP_n / n$ の演算は、除算ではなく単なるシフト演算で済ませることができる。このため、1画素単位の演算処理は、単純な加算器で実現できる。

【0052】なお、説明の簡略化のために、ディスプレイ上の水平方向 n 画素分に対し、テキストの補間係数を求めているが、ディスプレイ上の水平方向 n 画素垂直方向 m 画素分に対して、テキストの補間係数を求めることも可能である。

【0053】本発明では、除算処理は n 画素単位の処理となるため、描画ユニット6が n 画素処理し終わるまでの間に、除算が完了すればよい。このため、繰り返し除算器など、より簡易な構成で実現することができる。

【0054】また、高速化のため複数の描画ユニットを持つ構成では、一度に複数画素についてのZ, R, G, B, A, S, T, Q情報を計算する必要があるため、S, T, Qについて並列に線形補間するためのユニット1~4の他に、Z, R, G, B, Aについての並列演算を行うユニット、例えば色補間等の並列演算ユニットを設けるのが望ましい。

【0055】本発明は、以下のように作用する。

1) 描画すべき画素に対し、 n 画素先行した画素におけるテキスト座標を計算する。

2) 描画すべき画素(a)のテキスト座標と、 n 画素先行した画素(b)におけるテキスト座標を求め、2つのテキスト座標の差分から、画素(a)と画素(b)の間の画素についてのテキスト座標の線形補間係数を算出する。

3) 描画すべき画素のテキスト座標と、線形補間係数から、各々の画素におけるテキスト座標を線形補間で求める。

これにより、本発明では、3次元コンピュータグラフィックスのテキストチャマッピングにおいて、高速のテキストデータの座標算出を遠近感を損なわずに簡易に行うことができる。

【0058】また、本発明では、複数の描画ユニットを設け、これらユニットにより、複数の画素を同時に描画する構成において、テキスト座標の遠近感の補正を1つの共通ユニットにより行うようにすることができる。さらに、複数の描画ユニットに必要となる個別のテキスト座標を、共通ユニットの情報から同時に算出することもできる。

【0059】

【発明の実施の形態】以下、本発明の実施の形態を説明

する。図2は、本発明の第1の実施の形態に係る構成例であって、特に8画素単位で除算を行う場合の構成例を示す。

【0060】図2において、被除数・除数算出ユニット21、除算ユニット22、線形係数算出ユニット23、テクスチャ座標取得ユニット24は、それぞれ図1に示す被除数・除数算出ユニット1、除算ユニット2、線形係数算出ユニット3、テクスチャ座標取得ユニット4に対応する。

【0061】さらに、2次元のテクスチャデータを扱うために、水平方向と垂直方向の2つのテクスチャ座標を算出する必要があるため、テクスチャ水平座標処理ユニット20の他に、ユニット21~24と同等のユニットからなるテクスチャ垂直座標処理ユニット28を持つ。

【0062】Sel1~Sel4はセレクタ、Mul1、Mul2は乗算用シフト回路、R1~R9はレジスタ、Add1~Add3は加算器、Shift1、Shift2はシフト回路、Sub1、Sub2は減算器、Divは除算用シフト回路を表す。

【0063】被除数・除数算出ユニット21では、初期値T1、 ΔT 、Q1、 ΔQ を入力する。T1はセレクタSel1を通して、レジスタR1にセットされ、 ΔT は乗算用シフト回路Mul1で8倍されてレジスタR2にセットされる。 ΔT を8倍しているのは、8画素単位に処理するためであり、8は2の3乗の値であるため、シフト演算で済ませることができる。レジスタR1とR2の値は、加算器Add1で加算され、結果が除算ユニット22へ出力され、またセレクタSel1を通してレジスタR1へ再セットされる。以後、加算器Add1による加算が繰り返される。

【0064】これによって、「 $T_{n+1} = T_n + \Delta T * 8$ 」の演算が繰り返し実行されることになる。「 $Q_{n+1} = Q_n + \Delta Q * 8$ 」についても同様に演算が行われる。除算ユニット22では、被除数・除数算出ユニット21から入力した被除数 T_{n+1} を除数 Q_{n+1} で割る演算を行う。ここでは、それぞれシフト回路Shift1、Shift2によりビットの位置合わせを行った後、減算器Sub1による減算を繰り返し行って、商を求めている。シフト回路Shift3は結果の桁調整のためのものである。この結果、「 $TP_{n+1} = T_{n+1} / Q_{n+1}$ 」が求められる。

【0065】線形係数算出ユニット23は、除算ユニット22からの8画素単位で得られる除算結果 TP_{n+1} と、レジスタR7に保持していた1つ前の除算結果 TP_n とから、それらの差分 ΔTP_n を減算器Sub2により算出する。

【0066】テクスチャ座標取得ユニット24は、まずシフト回路で構成される除算器Divにより、 ΔTP_n を $1/8$ にし、除算結果 TP_n を TP の初期値として、これに加算器Add3により、 $\Delta TP_n / 8$ を足し込ん

でいく。これにより、8画素分のテクスチャ水平座標が求められることになる。この結果を順次、描画ユニット26に出力する。

【0067】テクスチャ垂直座標処理ユニット28も同様の処理を行う。図3は、本発明の第2の実施の形態に係る構成例を示す。図3の構成例は、描画の高速化のため8個の描画ユニット36-1~36-8を用いて、8画素分の描画を並列処理するものである。

【0068】図3において、被除数・除数算出ユニット31、除算ユニット32、線形係数算出ユニット33、テクスチャ座標取得ユニット34は、それぞれ図1に示す被除数・除数算出ユニット1、除算ユニット2、線形係数算出ユニット3、テクスチャ座標取得ユニット4に対応する。

【0069】図2に示す例とほぼ同様であるので、特に図2に示すものと違う部分について説明する。図3の例では、8画素分の描画を並列に実行するために、8個の描画ユニット36-1~36-8を持つ。また、Z、R、G、B、Aについての8並列演算を行う色補間等の並列演算ユニット39を持つ。

【0070】テクスチャ座標取得ユニット34は、テクスチャ座標 $TP_n (=Tp)$ と、テクスチャ座標値の線形補間係数 $\Delta TP_n / 8$ を0倍から7倍した値を並列で足し込むことで、隣り合う8画素分のテクスチャ座標を同時に求める。これにより、8画素分の描画ユニット36-1~36-8は即座に処理を開始することができるようになっている。

【0071】除算ユニット32が、図2に示す構成例での除算ユニット22と異なるのは、図2に示す構成例の場合には、1つの描画ユニット26により8画素の描画を行っている間に、次の除算結果を求めればよかったのに対し、図3では、8個の描画ユニット36-1~36-8で一度に8画素の描画処理を行うため、1画素分の処理時間で、除算結果を求める必要があるからである。このため、除算ユニット32は、パイプライン型の高速除算器になっている。

【0072】また、一度に複数画素についてのZ、R、G、B、A、S、T、Q情報を計算する必要があるため、S、T、Qについて並列に線形補間するためのユニットの他に、Z、R、G、B、Aについての並列演算を行う色補間等の並列演算ユニット39が設けられている。

【0073】

【発明の効果】以上説明したように、本発明によれば、テクスチャマッピング技術において、よりリアリティの高い画像を生成する上で必要となる遠近感の補正を簡易に行うことができるようになる。

【0074】図4は、図2に示す構成例で実施した描画結果を示す図である。視線との傾きが大きくなった図4(D)で、遠景部に若干の歪みが出ている。図4(E)

は、(D)に示す一部分を拡大したものであるが、歪みは1画素以下である。また、この例の市松模様のように境界が明確な図形では、歪みを認知できるが、図5に示す画像のように自然画をテクスチャデータとして使用した場合には、歪みを視覚的に捉えることはない。

【0075】このように、本発明によれば、画素ごとに除算した場合と比べても、遜色ない結果を、より簡易な構成で実現できる。さらに、高速化のために複数の描画ユニットを用いた並列構成をとる場合でも、除算処理部分は共通で、個々のプロセッサに与えるべきテクスチャの座標値を線形補間する部分だけ描画ユニット数に応じて用意すればよい。また、さらにコストの低減に貢献する。

【図面の簡単な説明】

【図1】本発明の原理説明図である。

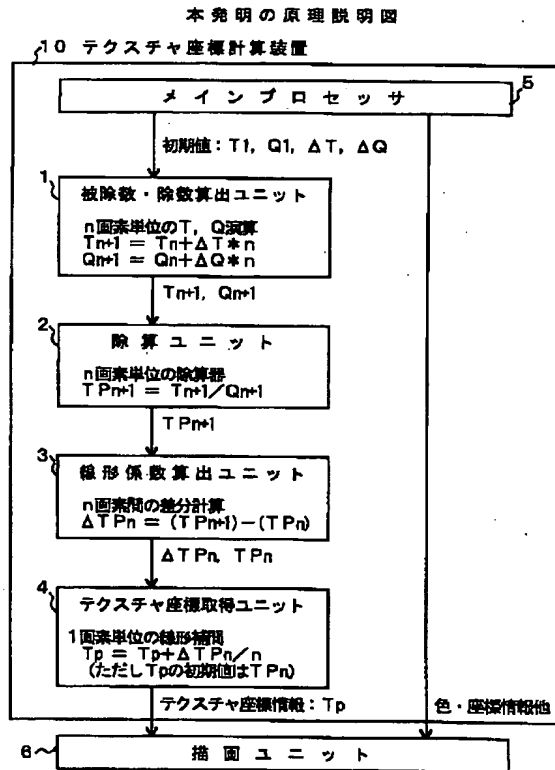
【図2】本発明の第1の実施の形態に係る構成例を示す図である。

【図3】本発明の第2の実施の形態に係る構成例を示す図である。

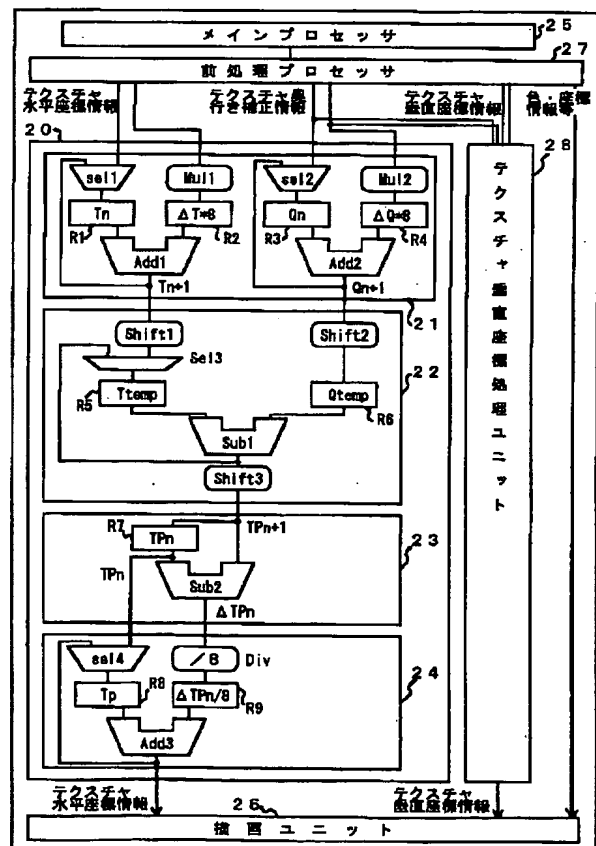
【図4】本発明の第1の実施の形態で実施した描画結果を示す図である。

【図5】自然画をテクスチャデータとして使用した場合*

【図1】



【図2】



*の描画結果を示す図である。

【図6】3次元コンピュータグラフィックスの一般的なシステムの全体構成を示す図である。

【図7】ポリゴンデータの例を示す図である。

【図8】ポリゴンデータのブレイクダウンで得る各データを示す図である。

【図9】3次元コンピュータグラフィックスにおける遠近感の表現例を示す図である。

【図10】3次元コンピュータグラフィックスにおける遠近感の表現とテクスチャの関係を説明する図である。

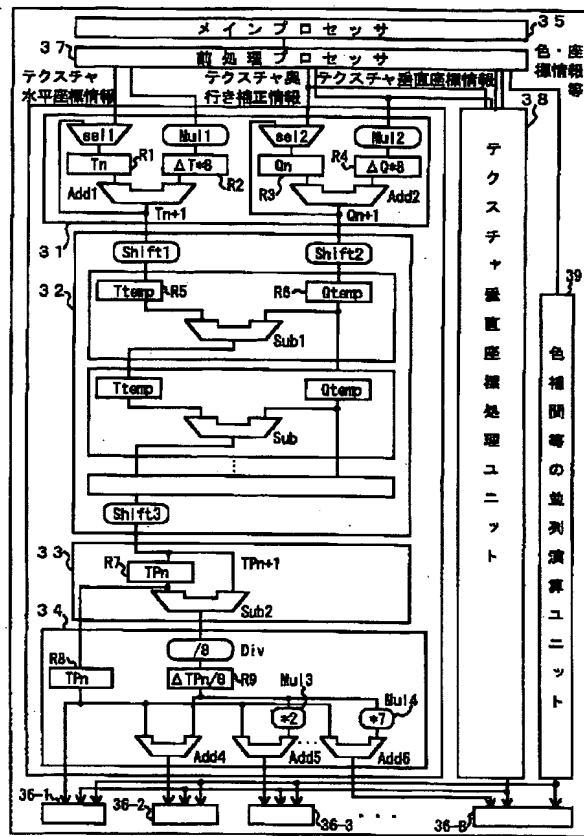
【図11】ポリゴンデータとテクスチャデータの対応を説明する図である。

【図12】テクスチャの座標補正を行わない場合の遠近感の表現例を示す図である。

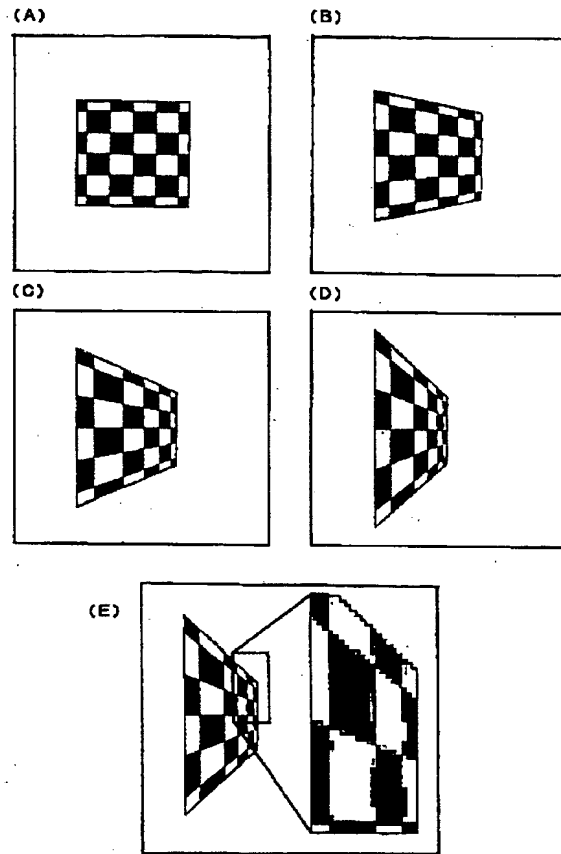
【符号の説明】

- 1 被除数・除数算出ユニット
- 2 除算ユニット
- 3 線形係数算出ユニット
- 4 テクスチャ座標取得ユニット
- 5 メインプロセッサ
- 6 描画ユニット
- 10 テクスチャ座標計算装置

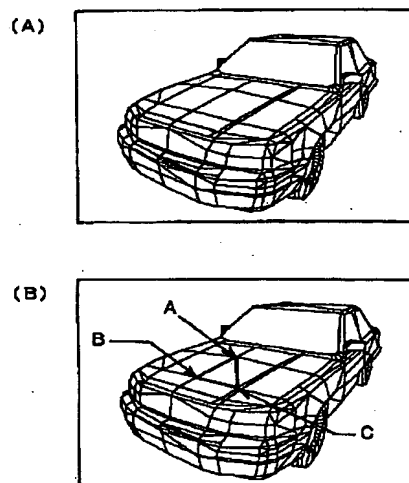
【図3】



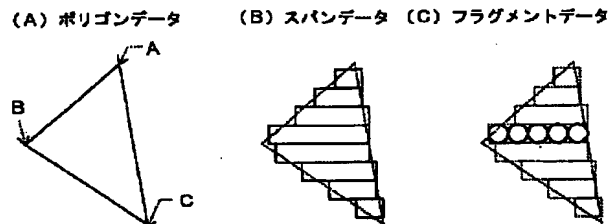
【図4】



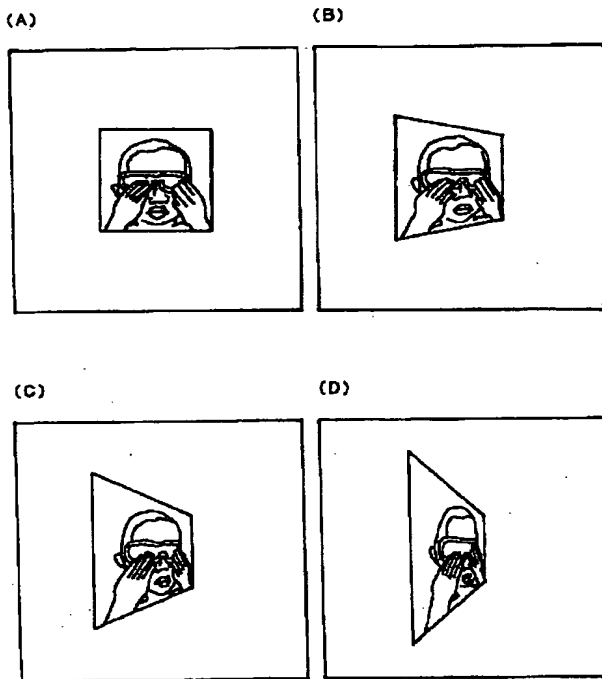
【図7】



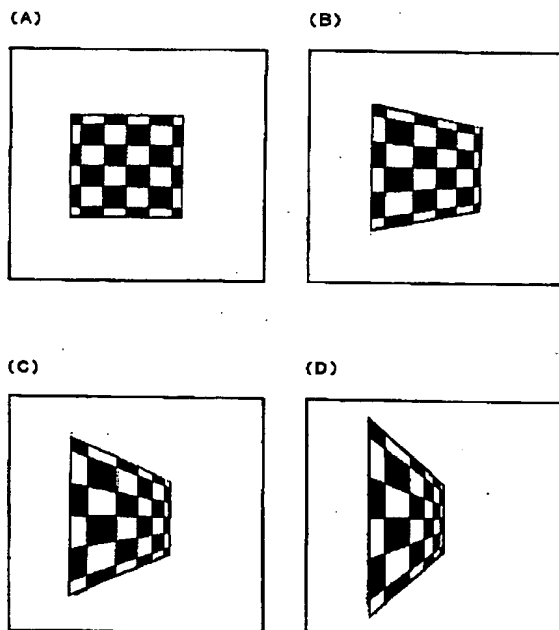
【図8】



【図5】

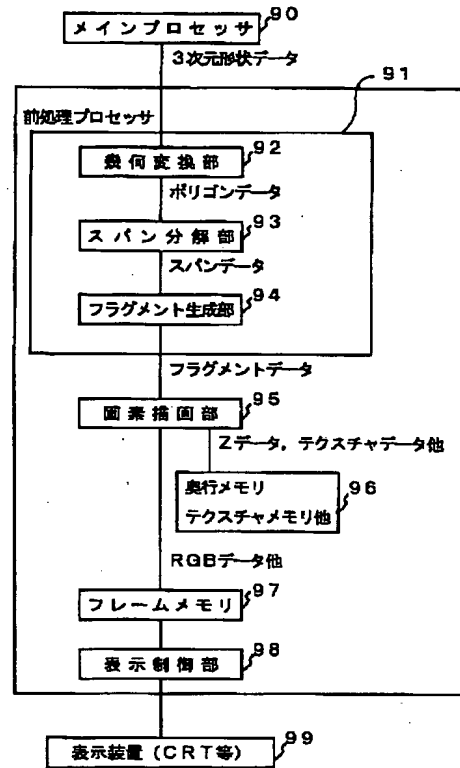


【図9】

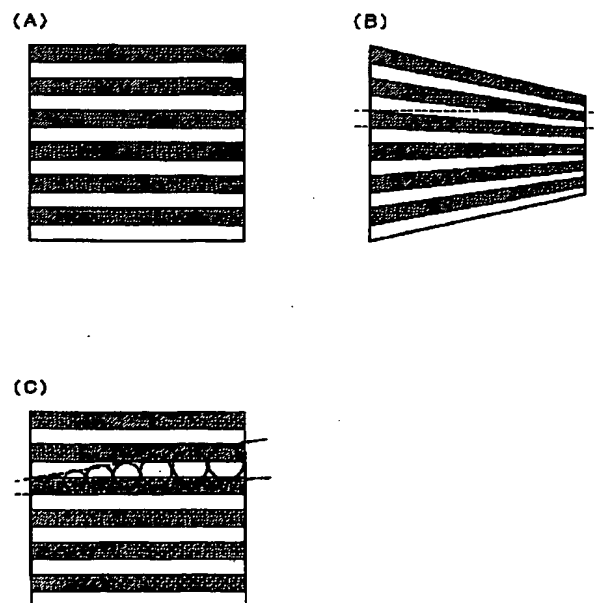


【図6】

3次元コンピュータグラフィックスの一般的なシステムの全体構成



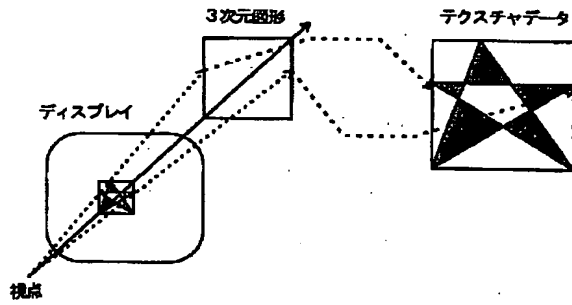
【図11】



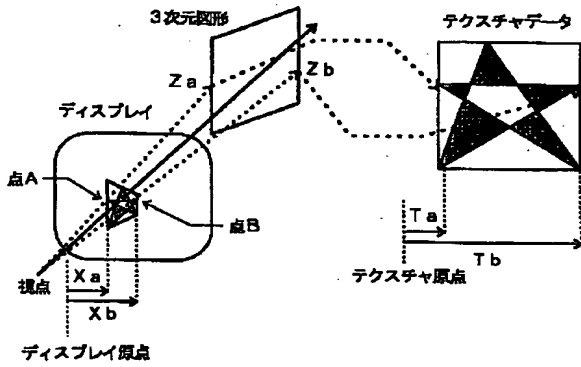
【図10】

3次元コンピュータグラフィックスにおける
遠近感の表現とテクスチャの関係の説明図

(A) 3次元図形が視線と直交する場合

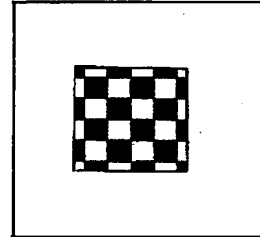


(B) 3次元図形が視線と直交しない場合

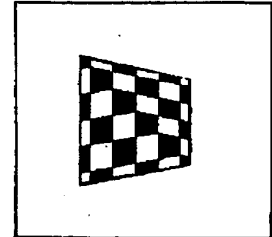


【図12】

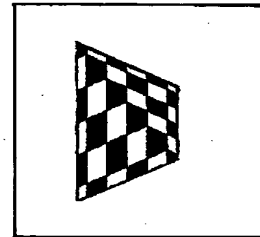
(A)



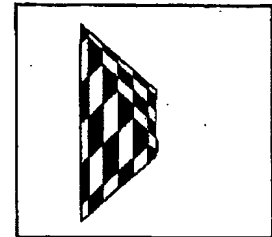
(B)



(C)



(D)



PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-185052

(43)Date of publication of application : 09.07.1999

(51)Int.Cl.

G06T 15/00

G06T 11/00

(21)Application number : 09-355995

(71)Applicant : FUJITSU LTD

(22)Date of filing : 25.12.1997

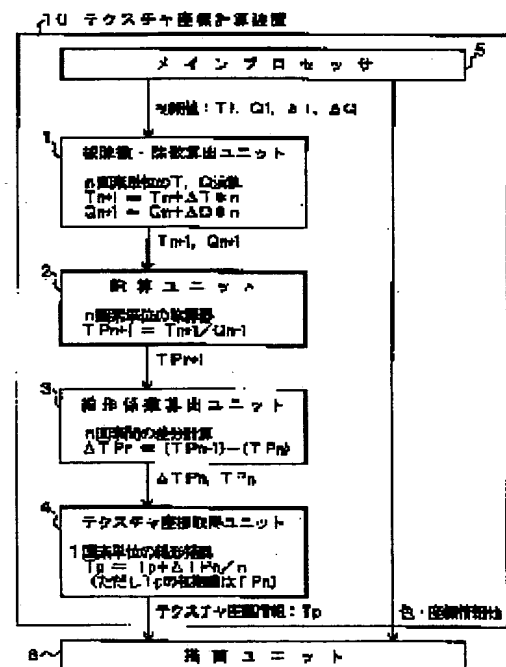
(72)Inventor : OOTSUKA TATSUSHI

(54) DEVICE AND METHOD FOR CALCULATING TEXTURE MAPPING COORDINATE OF THREE-DIMENSIONAL COMPUTER GRAPHICS

(57)Abstract:

PROBLEM TO BE SOLVED: To reduce the amount of a division processing to become required for correction of distortion of depth in sticking texture data regarding a device and method for calculating a texture mapping coordinate of three-dimensional computer graphics.

SOLUTION: A dividend and a divisor to calculate a texture coordinate of a pixel (b) preceding a pixel (a) by n pixels are calculated by a dividend and divisor calculating unit 1 and the texture coordinate of the pixel (b) is calculated from values of the dividend and divisor by division by a divider unit 2. A linear interpolation coefficient is calculated from difference between the texture coordinates of the pixels (a), (b) by a linear coefficient calculating unit 3. The texture coordinates of respective pixels between the pixels (a) and (b) are calculated from the texture coordinate and the linear interpolation coefficient of the pixel (a) by a texture coordinate acquisition unit 4.



11-185052

* NOTICES *

JPO and INPIT are not responsible for any

5 damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.

2. **** shows the word which can not be translated.

10 3. In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

15 [Claim 1] A pixel (a) which is a device which computes texture-mapping coordinates of three-dimensional computer graphics, and should draw A texture coordinate, Said pixel (a) which should draw A pixel (b) which received and carried out n pixel precedence A texture coordinate which can be set is searched for and it is difference of two texture coordinates to said pixel (a). Said pixel (b) A linear interpolation coefficient of a texture coordinate about a pixel of a between. 1st means to compute, and said pixel (a) which should draw. A ** texture coordinate and said linear interpolation coefficient to said pixel (a) Said pixel (b) A texture-mapping coordinates computing device of three-dimensional computer graphics provided with 2nd means to search for a texture coordinate in each pixel of a between.

25 [Claim 2] To claim 1, are a texture-mapping coordinates computing device of three-dimensional computer graphics to indicate, and said 1st means, Said pixel (b) which carried out n pixel precedence A dividend and a divisor calculation part which computes a dividend and a divisor of n pixel unit for searching for a texture coordinate by division, Division is done with a calculated value and it is said pixel (b). A division part which computes a texture coordinate, said pixel (a) A texture coordinate and said pixel (b). Difference with a ** texture coordinate to said pixel (a) Said pixel (b) Comprising a linear coefficient calculation part which computes a linear interpolation coefficient of a texture coordinate about a pixel of a between, said 2nd means is said pixel (a). To a texture coordinate, said linear interpolation coefficient. It adds and they are said pixel (a) and said pixel (b). A texture-mapping coordinates computing device of three-dimensional computer graphics comprising a texture coordinate acquisition part which searches for a texture coordinate in each pixel of a between.

35 [Claim 3] To claim 2, are a texture-mapping coordinates computing device of

three-dimensional computer graphics to indicate, and said texture coordinate acquisition part, Said pixel (a) Said linear interpolation coefficient is made a texture coordinate k times (k= 0, 1 and 2, --, n-1), respectively. It has a means to perform an operation to add in parallel, and is said pixel (a). Said pixel (b) A texture-mapping coordinates computing device of three-dimensional computer graphics computing a texture coordinate of each pixel of a between simultaneously.

[Claim 4]A pixel (a) which is the method of computing texture-mapping coordinates of three-dimensional computer graphics, and should draw A texture coordinate, Said pixel (a) which should draw A pixel (b) which received and carried out n pixel precedence A texture coordinate which can be set is searched for and it is difference of two texture coordinates to said pixel (a). Said pixel (b) A linear interpolation coefficient of a texture coordinate about a pixel of a between. It computes and is said pixel (a) which should draw. A texture coordinate and said linear interpolation coefficient to said pixel (a) Said pixel (b) A texture-mapping coordinates calculation method of three-dimensional computer graphics searching for a texture coordinate in each pixel of a between.

* NOTICES *

JPO and INPIT are not responsible for any

damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention]This invention relates to the art of realizing high-speed drawing of three-dimensional computer graphics, and relates to a device which calculates the texture coordinate which is needed for especially texture mapping of three-dimensional computer graphics at high speed by comparatively simple

composition, and a method for the same.

[0002]In recent years, three-dimensional computer graphics does not remain in visualization of the computer design which is the conventional main use, but has come to be mostly used in the field of a game, etc. In these fields, in order to react in real time to a user's input, the throughput of high-speed three-dimensional computer graphics is needed.

[0003]For the purpose, little formed data needs to generate the high picture of a reality, and the art called texture mapping is used. Texture mapping is processing which prepares the two-dimensional picture beforehand called texture data, and makes the pixel unit at the time of drawing of a polygon reflect texture data in the pixel information of a polygon with reference to these texture data. When few polygons also stick complicated texture data by this, the high picture of a reality is generable.

[0004]

[Description of the Prior Art]First, the entire configuration of the common system of three-dimensional computer graphics is explained.

[0005]Drawing 6 is a figure showing the entire configuration of the common system of three-dimensional computer graphics. The general purpose processor 90 is a processor which performs actual application of three-dimensional computer graphics, and three-dimensional information is managed. For example, in mechanical-system CAD, the data of the shape of a building, etc. is managed for the data of the shape of the body of a car, etc. as three-dimensional information at CAD of the architecture system.

[0006]Since these data is only mere coordinate data rows etc. on memory storage, That which makes like a car the picture that human being looks at thing based on these data, like a building if it is a building if an object is a car (it visualizes) is three-dimensional computer graphics processing (three-dimensional CG processing).

[0007]Although the content-type of the three-dimensional information which the application of the general purpose processor 90 manages is generally various, in order that it may make composition of three-dimensional CG processing easy with each application, the three-dimension shape data which three-dimensional CG processing receives is patternized to some extent.

[0008]As one of them, there is the technique of approximating and expressing three-dimensional information as shown in drawing 7 with two or more polyhedrons. As shown in drawing 7 (A), the body of a car can be actually approximated and expressed in a polygonal combination. Although a rectangular combination is expressing the body of a car in this example, it is possible to take various modes of expression by a system. Generally, since a rectangle and a still more complicated polygon can also be expressed combining two or more triangles, there are many systems expressed combining a triangle.

[0009]Hereafter, expression which uses a triangle for simplification of explanation is explained. A triangle has a vertex of the point A, B, and C, as shown in drawing 7 (B), and in three-dimensional computer graphics, information, including the

three-dimensional coordinates in these three vertices, a color, a normal, a texture coordinate, etc., is passed to the geometric converter 92 as three-dimension shape data.

[0010]In addition to these information, the geometric converter 92 computes the information called the following polygon data aligned with the screen which should actually be displayed using a viewpoint, a light source, the construction material information on a field, etc.

[0011]1) Also when [which is the sexual desire news 3 texture-coordinate information on each peak by the illumination condition from depth information 2 light source for judging a context to be X and the Y coordinate on a screen, and realizing a hidden surface removal] performing processing of the pretreatment processor 91 by the general purpose processor 90, be.

[0012]Although polygon data are the information in each peak, it is necessary to carry out breakdown of them to the information for from now on smearing away the inside of a polygon. The Sepang decomposition part 93 and the fragmentation generation part 94 process this breakdown.

[0013]In the Sepang decomposition part 93, the polygon data which comprise the peak as shown in drawing 8 (A) are divided into the horizontal direction of a screen, and the Sepang data shown in drawing 8 (B) which consists of information on the horizontal drawing starting point, and increment and an end point is generated.

[0014]The fragmentation generation part 94 generates X and the Y coordinate of one pixel unit called the fragmentation data shown in drawing 8 (C), depth information, sexual desire news, texture information, etc. based on this Sepang data.

[0015]Based on this fragmentation data, the pixel drawing part 95 performs reading of texture-mapping data, calculation of sexual desire news, etc. from the judgment (hidden surface processing) of the context of the drawing point by depth information, and a texture coordinate, and determines the color of the pixel of a final drawing point, etc. Generally, in the case of computer graphics, the value which takes the initial of three-primary-colors red-green blue of a color, and is called an RGB value is written in the frame memory 97.

[0016]The value called an alpha value may be used as penetration information on a color, and it treats as a RGBA value in this case. The display control part 98 reads the value of this frame memory 97 about 60 times in 1 second, and displays it on the displays 99, such as CRT.

[0017]Since the three-dimension shape data which the pretreatment processor 91 treats includes many information, including a viewpoint, light source information besides vertex information, etc., and also it is various forms in a relation with application, it is omitted here and explains the data below polygon data without great difference by every system.

[0018]The polygon data shown in drawing 8 (A) have X and a Y coordinate, depth Z information, color RGBA information, the texture coordinate information S and T, and the information Q on the distortion correction of depth about the three peaks A, B, and

C.

[0019]These information on the peak A is expressed as follows.

"Xa, Ya, Za, Ra, Ga, Ba, Aa, Sa, Ta, Qa"

Similarly, about the peaks B and C, it expresses as follows, respectively.

5 [0020]

"Xb, Yb, Zb, Rb, Gb, Bb, Ab, Sb, Tb, Qb"

"Xc, Yc, Zc, Rc, Gc, Bc, Ac, Sc, Tc, Qc"

10 The Sepang data shown in drawing 8 (B) is the starting point in each horizon, the increment, and the information on an end point which decomposed horizontally the triangle which consists of the three vertices A, B, and C. X and the Y coordinate in the starting point, depth Z information, color RGBA information, the texture coordinate information S and T, the distortion correction information Q on depth, and the information on those increment are expressed as follows.

[0021]

15 "Xs, Ys, Zs, Rs, Gs, Bs, As, Ss, Ts, Qs"

"dXs, dYs, dZs, dRs, dGs, dBs, dAs, dSs, dTs, dQs"

As shown in drawing 8 (B), supposing the peak is located in a line, when the starting point is on the neighborhood AB, the Sepang information on Y coordinate Yi can be found in the following calculations.

20 [0022] $Ms(Yi) = (Mb - Ma) * (Yi - Ya) / (Yb - Ya)$

However, $Yi = Ya - Yb$, and M express each of X, Z, R, G, B, A, S, T, and Q.

[0023]Although division occurs by $(Yb - Ya)$ in a formula, this can be transposed to multiplication by not being based on change of Yi, but calculating reciprocal $1/(Yb - Ya)$ previously, since it is fixed.

25 [0024]Increment is calculable as follows.

What is necessary is just not to base $dMs = (Yb - Ya) * (Mc - Ma) - Mb (Yc - Ya) / (Yb - Ya) * (Xc - Xa) - Xb (Yc - Ya)$ increment on Yi, but to calculate it once first, since it is fixed. An end point requires only an X coordinate. If this is set to Xe, it is calculable as follows.

30 [0025] $Xe(Yi) = (Xc - Xa) * (Yi - Ya) / (Yc - Ya)$

The fragmentation data shown in drawing 8 (C) is the every 1 pixel information on Sepang, and can be computed from the starting point and increment.

35 [0026]X and the Y coordinate of each pixel, depth Z information, color RGBA information, the texture coordinate information S and T, the information Q on the distortion correction of depth, and the information on increment are expressed as follows.

[0027]"Xp, Yp, Zp, Rp, Gp, Bp, Ap, Sp, Tp, Qp"

If each information and increment of the starting point on Sepang Xs and Xe are set to Ms and dMs (respectively [Z, R, G, B, A, S, T, and Q]), the fragment information of X coordinate Xi on Sepang is calculable as follows. [M]

40 [0028] $Mf(Xi) = Ms + dMs * (Xi - Xs) / (Xe - Xs)$

However, $X_i = X_s - X_e$, and M express each of Z, R, G, B, A, S, T, and Q.

[0029] Thus, although the pixel drawing part 95 starts actual drawing processing with fragmentation data, if the texture coordinate Sf and Tf are applied to access of a texture memory as it is, since it has distortion in depth, it will become an unnatural picture. In order to avoid this, it is necessary to calculate Sf /Qf and Tf /Qf and to perform distortion correction of depth in the pixel drawing part 95.

[0030] Although distortion occurs strictly also about a RGBA value, since change of sexual desire news is smooth, distortion is hardly perceived and distortion correction can be omitted.

[0031] The distortion correction of this depth is explained in more detail according to an example. He is trying to draw small in three-dimensional computer graphics as it separates from a viewpoint, in order to express depth perception.

[0032] Drawing 9 sticks a checkered texture on a rectangle, and shows signs that rectangular vertical Chuo Line was rotated as a center. In the figure shown in drawing 9 (A), although the rectangle lies at right angles to a look, according to drawing 9 (B) - (D), a rectangle inclines gradually and is displayed. Thus, according to depth perception, this side is large, and the checker on a texture is also shrunken by a distant place, and needs to be expressed with three-dimensional computer graphics.

[0033] For this reason, as shown in drawing 10 (A), when drawing the polygon data which intersect perpendicularly with a look, the distance between the pixels on the texture data corresponding to an adjacent pixel is the same, but. As shown in drawing 10 (B), when drawing the polygon data which do not intersect perpendicularly with a look, since the distance between the pixels on the texture data corresponding to an adjacent pixel differs, by the near side (display surface left-hand side) and the back side (display surface right-hand side), amendment is needed.

[0034] Namely, in mapping to the rectangle which does not intersect perpendicularly with a look as shown in drawing 11 (B), texture data as shown, for example in drawing 11 (A). Since it spreads as the range of corresponding texture data is on a back side (display surface right-hand side), and shown in drawing 11 (C), amendment of the texture coordinate mapped by polygon data is needed.

[0035] In drawing 10 (B), the pixel coordinate of the point A and the point B X_a , X_b (originally, although the coordinates on a display are expressed with two dimensions, since they are simplification of explanation, they omit a Y coordinate and make it one element), When the coordinates on Z_a , Z_b , and texture data correspond like T_a and T_b in depth (let it be one element since it is simplification of explanation, although the texture coordinate of two-dimensional original is expressed with two dimensions), the texture coordinate T_p over the pixel of the coordinates X_p between the point A and the point B is expressed as follows.

[0036] It is necessary to perform <formula 1> $T_p = (T_1 + \Delta T * (X_p - X_a)) / (Q_1 + \Delta Q * (X_p - X_a))$ compensation process of these for every pixel between the point A and the point B.

[0037]It is here and they are $T1 = Ta / Za$ $T2 = Tb / Zb$ $\Delta T = (T2 - T1) / (Xb - Xa)$

$Q1 = 1/Za$ $Q2 = 1/Zb$ $\Delta Q = (Q2 - Q1) / (Xb - Xa)$

It comes out, and it is and these values are constant about the point between the point A and the point B.

[0038]

[Problem(s) to be Solved by the Invention]As shown in the above-mentioned <formula 1>, division is needed for processing which searches for the texture coordinate Tp . Usually, data processing is needed for dividing operation one by one. In being shorter than time until the time of the drawing processing per pixel searches for a divided result by sequential processing, it is necessary to pipeline a dividing operation part, and a hardware scale increases so that dividing operation may not serve as a bottleneck.

[0039]In order to perform parallel processing for every pixel using two or more drawing processors, the hardware of dividing operation is needed for each drawing processor. Since depth perception is not reflected in a texture on the other hand as shown in drawing 12 although there is also selection which omits the compensation process of a texture itself in a simple system, a picture becomes what has sense of incongruity unnatural expression of depth and large.

[0040]An object of this invention is to reduce the amount of dividing processes which is needed for the distortion correction of depth in attachment of texture data.

[0041]

[Means for Solving the Problem]In this invention, when acquiring a texture coordinate, division is done by n pixel unit instead of doing division by one pixel unit, and a texture coordinate value per pixel between the pixel is calculated by linear interpolation.

[0042]Conventionally, between $Xs - Xe$, to the fragmentation generation part 94 side shown in drawing 6, generate 1 pixel at a time, and in the pixel drawing part 95 for access to a texture memory. To having calculated S , T , and Q , by this invention, these processings are separated, and the fragmentation generation part 94 side computes two or more pixels, does division with a texture coordinate computing device explained below, asks for every 1 pixel between calculated pixels by linear interpolation, and passes it to a drawing unit (pixel drawing part). In a drawing unit, division is unnecessary any longer. Since it is so small that there is no influence of distortion or it can ignore about other Z , R , G , B , and A information, a value which it calculated 1 pixel at a time by the fragmentation generation part side is put into a drawing unit of a pixel.

[0043]Drawing 1 is a principle explanatory view of this invention. The texture coordinate computing device 10 consists of a dividend and the divisor calculation unit 1, the division unit 2, the linear coefficient calculation unit 3, the texture coordinate acquisition unit 4, and the general purpose processor 5. The drawing unit 6 accesses a texture memory in which texture data are stored based on a texture coordinate which

the texture coordinate computing device 10 computed, sticks texture data and draws.

[0044]Below, as it explained in drawing 10 (B), coordinates on Za, Zb, and texture data are explained [a pixel coordinate of the point A and the point B] for Xa, Xb, and depth as Ta and Tb.

5 [0045] $T1 = Ta / Za$ $T2 = Tb / Zb$ $\Delta T = (T2 - T1) / (Xb - Xa)$

$Q1 = 1/Za$ $Q2 = 1/Zb$ $\Delta Q = (Q2 - Q1) / (Xb - Xa)$

It carries out. Although expressed with two dimensions, since it is simplification of explanation, a texture coordinate of two-dimensional original is explained as one element.

10 [0046]A dividend and the divisor calculation unit 1 are units which compute a dividend and a divisor used in the division unit 2. A dividend and the divisor calculation unit 1 make T1 and Q1 an initial value, and adds a value which increased delta T and delta Q n times as a delta value. That is, the next operation is performed.

[0047]Here, values, such as T1, Q1, delta T, delta Q, calculate from the point A before the point B beforehand, before [which is $T_{n+1} = T_n + \Delta T * n$ $Q_{n+1} = Q_n + \Delta Q * n$] processing of a pixel in the meantime starts, since it is fixed. This calculation may be performed, for example by the general purpose processor 5, although not illustrated, a pretreatment processor is placed between a dividend and the divisor calculation unit 1, and it may be made to process in the general purpose processor 5 and drawing 1 here.

20 [0048]The division unit 2 is a unit which computes a texture coordinate for every n pixel which is dividend T_{n+1} and divisor Q_{n+1} to divided result TP_{n+1} which were computed with a dividend and the divisor calculation unit 1. The next operation is performed.

[0049] $TP_{n+1} = T_{n+1} / Q_{n+1}$ linear-coefficient calculation unit 3 is units which compute difference ΔTP_n of them from divided result TP_{n+1} obtained by n pixel unit, and the divided result TP_n in front of one. The next operation is performed.

[0050] $\Delta TP_n = (TP_{n+1}) - (TP_n)$

The texture coordinate acquisition unit 4 is a unit which calculates the texture coordinate TP of one pixel unit. The texture coordinate acquisition unit 4 makes the divided result TP_n an initial value, and adds $\Delta TP_n / n$ as a linear interpolation coefficient. That is, the next operation is performed.

30 [0051] $TP = TP + \Delta TP_n / n$ (however, an initial value of TP TP_n)

An operation of $\Delta TP_n / n$ can be substituted for mere shift operation instead of division by considering n as involution of 2. For this reason, data processing of one pixel unit is realizable with a simple adding machine.

[0052]Although an interpolation coefficient of a texture is calculated from a part for a horizontal direction n pixel on a display for simplification of explanation, it is also possible to calculate an interpolation coefficient of a texture from a part for a horizontal direction n pixel perpendicular direction m pixel on a display.

40 [0053]In this invention, since it becomes processing of n pixel unit, by the time the drawing unit 6 finishes carrying out n pixel processing, division should just complete a

dividing process. For this reason, it is realizable with simpler composition, such as a repetition divider.

[0054]In composition with a drawing unit of plurality for improvement in the speed. Since it is necessary to calculate Z about two or more pixels, R, G, B, A, S, T, and Q information at once, It is desirable to provide parallel operation units, such as a unit which performs parallel operation about Z, R, G, B, and A other than the units 1-4 for carrying out linear interpolation in parallel about S, T, and Q, for example, color interpolation etc.

[0055]This invention acts as follows.

1) Calculate a texture coordinate in a pixel which carried out n pixel precedence to a pixel which should draw.

[0056]2) Pixel (a) which should draw A texture coordinate and pixel (b) which carried out n pixel precedence A texture coordinate which can be set is searched for and it is difference of two texture coordinates to a pixel (a). Pixel (b) A linear interpolation coefficient of a texture coordinate about a pixel of a between is computed.

[0057]3) Search for a texture coordinate of a pixel which should draw, and a texture coordinate in each pixel from a linear interpolation coefficient by linear interpolation. Thereby, in this invention, coordinate computation of high-speed texture data can be simply performed in texture mapping of three-dimensional computer graphics, without spoiling depth perception.

[0058]In this invention, two or more drawing units are provided and one common unit can amend depth perception of a texture coordinate in composition which draws two or more pixels simultaneously with these units. An individual texture coordinate which is needed for two or more drawing units is also simultaneously computable from information on a common unit.

[0059]

[Embodiment of the Invention]Hereafter, an embodiment of the invention is described. Drawing 2 is an example of composition concerning a 1st embodiment of this invention, and shows the example of composition in the case of doing division especially by eight pixel units.

[0060]In drawing 2, a dividend and the divisor calculation unit 21, the division unit 22, the linear coefficient calculation unit 23, and the texture coordinate acquisition unit 24 are equivalent to the dividend and the divisor calculation unit 1, the division unit 2, the linear coefficient calculation unit 3, and the texture coordinate acquisition unit 4 which are shown in drawing 1, respectively.

[0061]Since it is necessary to compute two texture coordinates, a horizontal direction and a perpendicular direction, in order to treat two-dimensional texture data, it has the texture vertical coordinate handling unit 28 which consists of a unit equivalent to the units 21-24 other than the texture horizontal-coordinates handling unit 20.

[0062]Sel1-Sel4 -- a selector, Mul1, and Mul2 -- Add1-Add3, an adding machine, Shift1, and Shift2 express a register, a shift circuit, Sub1, and Sub2 express a subtractor, and,

as for the shift circuit for multiplication, and R1-R9, Div expresses the shift circuit for division.

[0063]In a dividend and the divisor calculation unit 21, the initial values T1 and delta T, Q1, and deltaQ are inputted. T1 lets selector Sel1 pass and it is set in the register R1, and deltaT is increased 8 times by shift circuit Mul1 for multiplication, and is set in the register R2. deltaT is increased 8 times in order to process to eight pixel units, and since 8 is a value of the cube of 2, it can be substituted for shift operation. It is added by adding machine Add1, and a result is outputted to the division unit 22, and re-setting of the value of the registers R1 and R2 is carried out through selector Sel1 to the register R1. Henceforth, addition by adding machine Add1 is repeated.

[0064]Repeat execution of the operation of " $T_{n+1} = T_n + \Delta T * 8$ " will be carried out by this. An operation is similarly performed about " $Q_{n+1} = Q_n + \Delta Q * 8$." In the division unit 22, the operation which divides dividend T_{n+1} inputted from the dividend and the divisor calculation unit 21 by divisor Q_{n+1} is performed. Here, after shift circuit Shift1 and Shift2 perform alignment of a bit, respectively, it carries out by repeating subtraction by subtractor Sub1, and is asking for the quotient. Shift circuit Shift3 is a thing for beam adjustment of a result. As a result, " $TP_{n+1} = T_{n+1} / Q_{n+1}$ " are calculated.

[0065]The linear coefficient calculation unit 23 computes difference ΔTP_n of them by subtractor Sub2 from divided result TP_{n+1} obtained by eight pixel units from the division unit 22, and the divided result TP_n before [one] holding to the register R7.

[0066]The texture coordinate acquisition unit 24 sets ΔTP_n to one eighth with the divider Div which comprises a shift circuit first, and adds $\Delta TP_n / 8$ for the divided result TP_n to this by adding machine Add3 as an initial value of T_p . By this, the texture horizontal coordinates for 8 pixels will be searched for. This result is outputted to the drawing unit 26 one by one.

[0067]Processing that the texture vertical coordinate handling unit 28 is also the same is performed. Drawing 3 shows the example of composition concerning a 2nd embodiment of this invention. The example of composition of drawing 3 carries out parallel processing of the drawing for 8 pixels using the eight drawing units 36-1 to 36-8 for improvement in the speed of drawing.

[0068]In drawing 3, a dividend and the divisor calculation unit 31, the division unit 32, the linear coefficient calculation unit 33, and the texture coordinate acquisition unit 34 are equivalent to the dividend and the divisor calculation unit 1, the division unit 2, the linear coefficient calculation unit 3, and the texture coordinate acquisition unit 4 which are shown in drawing 1, respectively.

[0069]Since it is the same as that of the example shown in drawing 2 almost, the portion different from what is shown especially in drawing 2 is explained. In the example of drawing 3, in order to perform drawing for 8 pixels in parallel, it has the eight drawing units 36-1 to 36-8. It has the parallel operation units 39, such as color interpolation which performs 8 parallel operation about Z, R, G, B, and A.

[0070]The texture coordinate acquisition unit 34 searches for simultaneously the texture coordinate for 8 pixels which are it being parallel and adding the value which increased linear interpolation coefficient $\Delta TP_n / 8$ of the texture coordinate value 7 times from 0 time, and adjoin the texture coordinate $TP_n (=TP)$. Thereby, the drawing unit 36-1 to 36-8 for 8 pixels can start processing now immediately.

[0071]That the division unit 32 differs from the division unit 22 in the example of composition shown in drawing 2, In the case of the example of composition shown in drawing 2, while the one drawing unit 26 is performing 8-pixel drawing, the following divided result to what was necessary having been just to have asked in drawing 3. In order for the eight drawing units 36-1 to 36-8 to perform 8-pixel drawing processing at once, it is the processing time for 1 pixel, and is because it is necessary to search for a divided result. For this reason, the division unit 32 is a pipeline type high-speed divider.

[0072]Since it is necessary to calculate Z about two or more pixels, R, G, B, A, S, T, and Q information at once, the parallel operation units 39, such as color interpolation which performs parallel operation about Z, R, G, B, and A other than the unit for carrying out linear interpolation in parallel about S, T, and Q, are formed.

[0073]

[Effect of the Invention]As explained above, according to this invention, in texture-mapping art, the depth perception which is needed when generating a picture with a higher reality can be simply amended now.

[0074]Drawing 4 is a figure showing the drawing result carried out in the example of composition shown in drawing 2. By drawing 4 (D) to which the inclination by a look became large, some distortion has appeared in the distant view part. Although drawing 4 (E) expands the part shown in (D), distortion is 1 pixel or less. Although a boundary can recognize distortion in a clear figure like the checker of this example, when natural drawing is used as texture data like the picture shown in drawing 5, distortion is not caught visually.

[0075]Thus, according to this invention, even if compared with the case where division is done for every pixel, the result as for which inferiority is not is realizable with simpler composition. Even when taking the parallel form using two or more drawing units for improvement in the speed, a dividing process portion is common, and in order that only the portion which carries out linear interpolation may prepare the coordinate value of the texture which should be given to each processor according to the number of drawing units, it contributes to reduction of cost further.

* NOTICES *

JPO and INPIT are not responsible for any

damages caused by the use of this translation.

5

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

10

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1]It is a principle explanatory view of this invention.

15 [Drawing 2]It is a figure showing the example of composition concerning a 1st embodiment of this invention.

[Drawing 3]It is a figure showing the example of composition concerning a 2nd embodiment of this invention.

20 [Drawing 4]It is a figure showing the drawing result carried out by a 1st embodiment of this invention.

[Drawing 5]It is a figure showing the drawing result at the time of using natural drawing as texture data.

[Drawing 6]It is a figure showing the entire configuration of the common system of three-dimensional computer graphics.

25 [Drawing 7]It is a figure showing the example of polygon data.

[Drawing 8]It is a figure showing each data obtained by the breakdown of polygon data.

[Drawing 9]It is a figure showing the example of expression of the depth perception in three-dimensional computer graphics.

30 [Drawing 10]It is a figure explaining the expression of depth perception and the relation of a texture to three-dimensional computer graphics.

[Drawing 11]It is a figure explaining correspondence of polygon data and texture data.

[Drawing 12]It is a figure showing the example of expression of the depth perception when not performing coordinate correction of a texture.

35 [Description of Notations]

1 A dividend and a divisor calculation unit

2 Division unit

3 Linear coefficient calculation unit

4 Texture coordinate acquisition unit
5 General purpose processor
6 Drawing unit
10 Texture coordinate computing device

5

[Translation done.]